

Máster en Programación avanzada en Python para Hacking, BigData y Machine Learning

Fundamentos de IA y Machine Learning

LECCIÓN 02

Entrenamiento y Evaluación

ÍNDICE

- ✓ Introducción
- ✓ Técnicas de evaluación
- ✓ Sobreentrenamiento e infraentrenamiento
- ✓ Métricas de evaluación

INTRODUCCIÓN

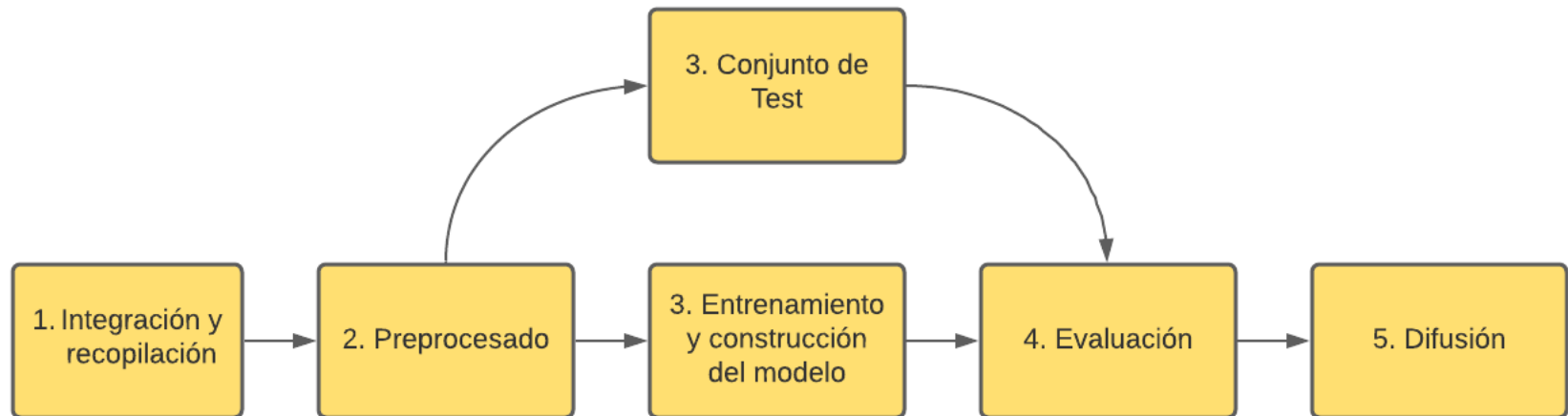
Cuando nos enfrentamos a un problema de aprendizaje automático, tratamos de generar un modelo basado en la recopilación de los datos para generar nuevo conocimiento. Para ello, las etapas de selección de la técnica, entrenamiento y evaluación de esta e interpretación de resultados se convierte en otras de las tareas esenciales del *machine learning*.

OBJETIVOS

Al finalizar esta lección serás capaz de:

- 1 Diferenciar entre modelo y algoritmo.
- 2 Dividir la base de datos para un correcto entrenamiento del modelo (ajuste de sus parámetros).
- 3 Evitar el sobreentrenamiento e infraentrenamiento.
- 4 Interpretar las matrices de confusión.
- 5 Evaluar un modelo a partir de las distintas métricas para cada tipo de problema.

1.1. Contextualización de la lección



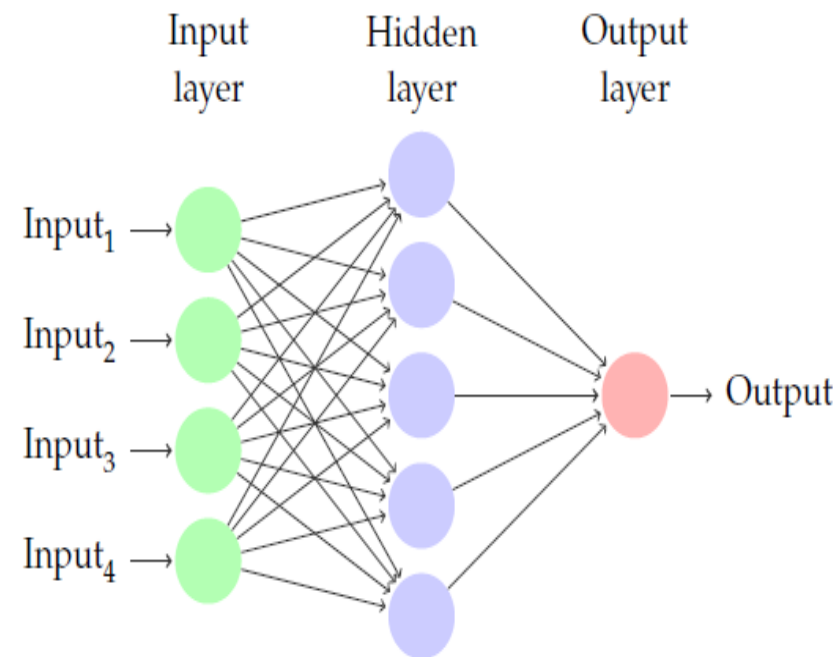
1.1. Contextualización de la lección

Tras la obtención de los datos y su importante fase de preprocesado, es el momento de tomar una serie de decisiones.

- **Determinar el tipo de conocimiento:** predictivo o descriptivo.
- **Analizar el tipo de técnica más apropiada:** clasificación, regresión, agrupación o *clustering*, reglas de asociación, sistemas de recomendación, ...
- **Dentro de la técnica, que modelo es el más interesante:** por ejemplo, en clasificación tendríamos regresión logística, árboles de decisión o redes neuronales, entre otros.
- **Establecer el algoritmo de aprendizaje más adecuado:** descenso por gradiente, algoritmos analíticos, algoritmos bioinspirados, ...

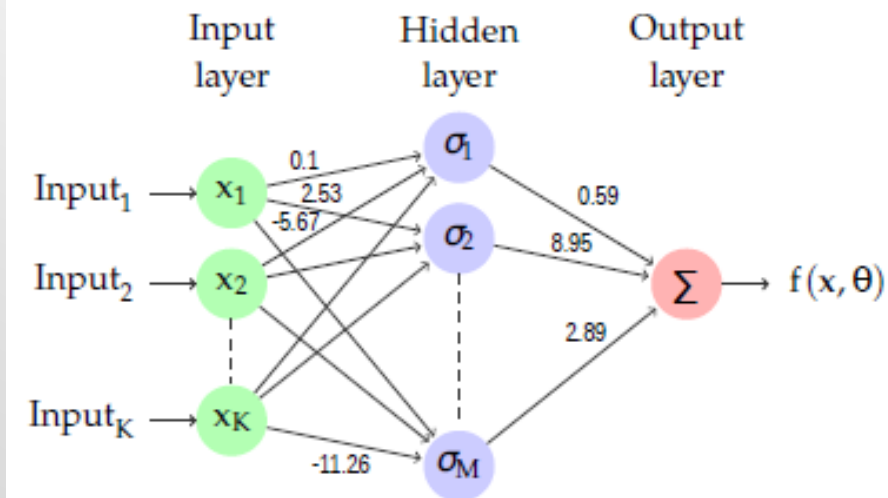
1.2. Modelo y algoritmo

- **Modelo:** un modelo es, en general, una función o estructura, que representa el conocimiento subyacente en un conjunto de datos. Dependiendo del objetivo del problema, tipo de variables de entrada, tipo de salida esperada, etc. habrá modelos más o menos apropiados. Un ejemplo sería la representación de una red neuronal, que trataremos con más detalle en la lección 4 de esta asignatura.



1.2. Modelo y algoritmo

- **Algoritmo:** un algoritmo es una secuencia de pasos con un fin. De este modo, en el contexto del aprendizaje automático, un algoritmo de aprendizaje se encarga de que el modelo aprenda de los datos, o expresado de otra forma, de que el modelo se ajuste a los datos. Así, en el ejemplo anterior de la red neuronal el algoritmo trataría de encontrar los pesos de la red para hacer que la variable predicha sea lo más similar posible a la variable objetivo real.



Modelo de red neuronal
entrenado

1.3. Entrenamiento y evaluación

- **Fase de entrenamiento:** se aplica un algoritmo que ajuste los parámetros del modelo al conjunto de datos.
- **Parámetros a estimar:** el conjunto de parámetros a estimar es distinto de un modelo a otro, sin embargo, existen una serie de nociones que son comunes en el proceso.
- **En aprendizaje supervisado:** ajustar los parámetros de forma que la variable objetivo y la real sean lo más parecidas, en otras palabras, el error sea el menor posible.

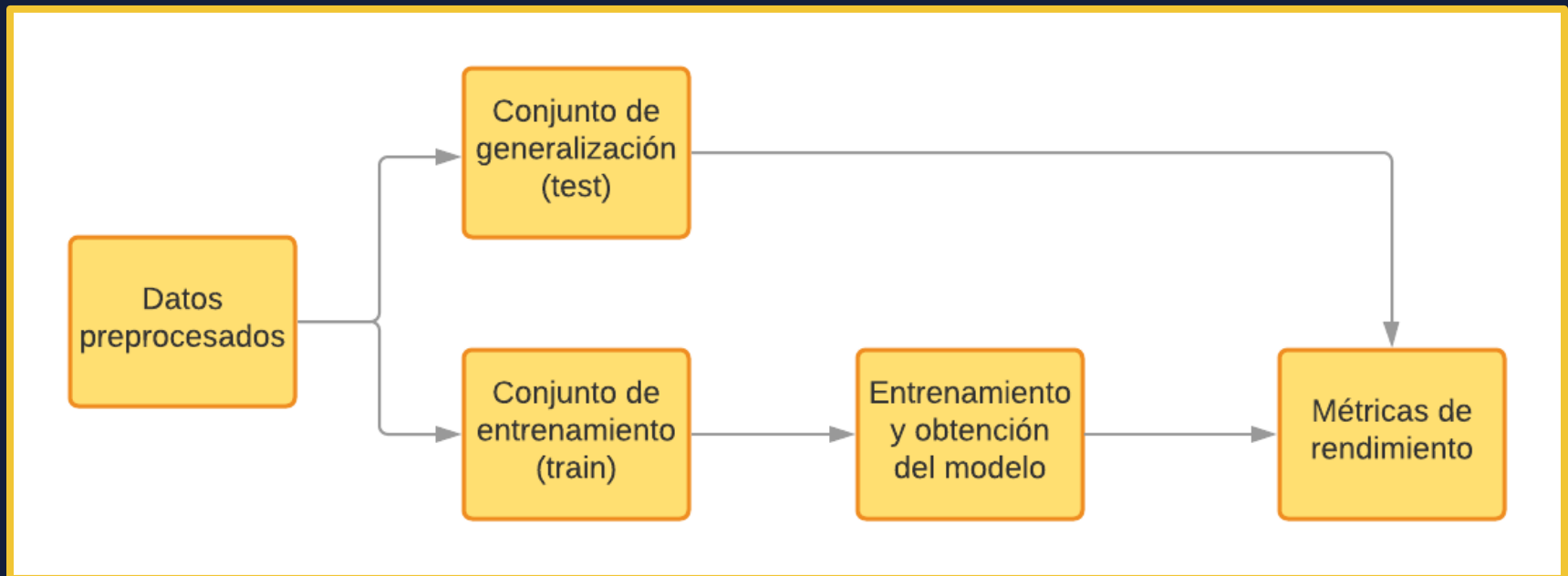
¿ES JUSTO ENTRENAR EL MODELO Y VALIDARLO SOBRE EL MISMO CONJUNTO DE DATOS?

2. Técnicas de evaluación

La evaluación de nuestros modelos debe ser realizada utilizando conjunto de datos que **no han sido usados durante el entrenamiento**.



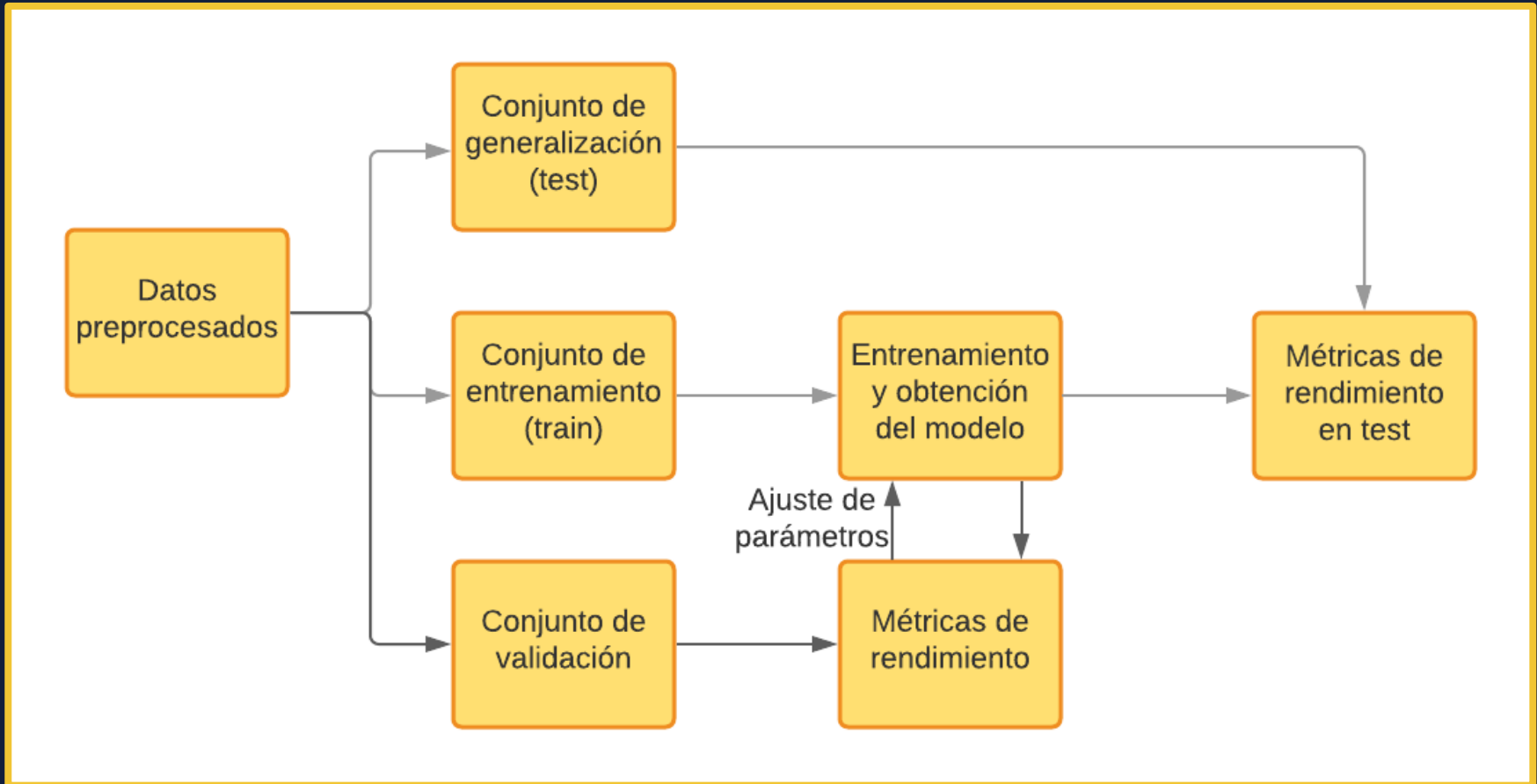
2.1. División en un conjunto de *train* y otro de *test*



2.1. División en un conjunto de *train* y otro de *test*

- Se debe seleccionar el porcentaje de entrenamiento y de generalización. Normalmente: 67%-33% ó 75%-25%.
- Ideal para conjuntos muy grandes → muestra representativa.
- Indicado para algoritmos lentos → procedimiento muy rápido.
- Dependen de la división inicial → gran varianza.

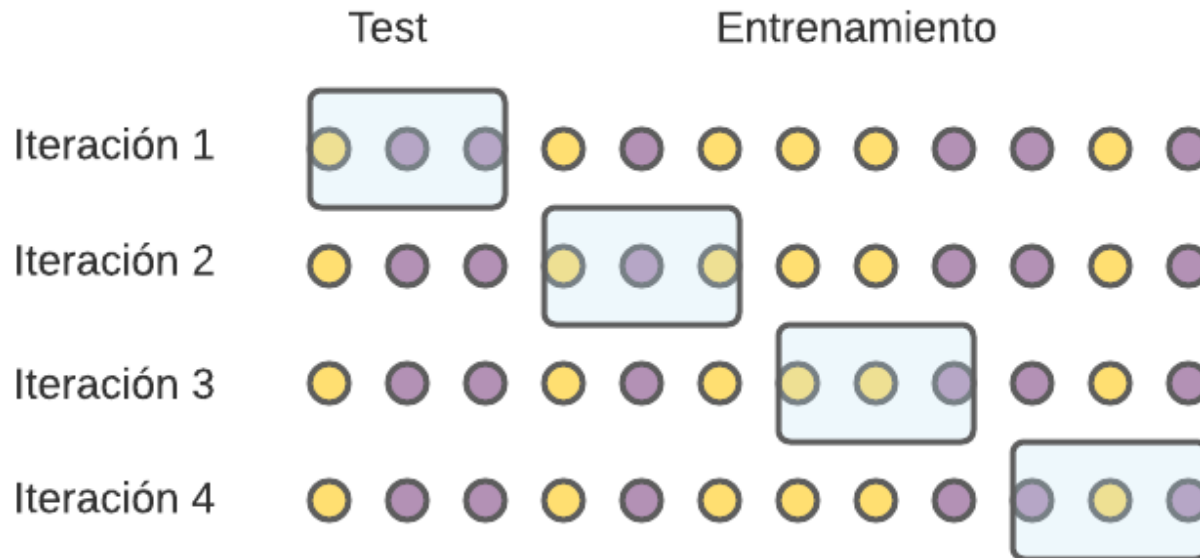
2.1. División en un conjunto de *train* y otro de *test* (conjunto de validación)



2.2. Repetición de varias divisiones aleatorias *Train-Test*

Una solución para minimizar la varianza del modelo obtenido es repetir el proceso anterior varias veces. De forma, tendremos tantos resultados de rendimiento como divisiones train-test se hagan. La idea es obtener el resultado medio y calcular su desviación típica.

2.3. Validación cruzada *k-fold*



2.3. Validación cruzada *k-fold*

- Menor varianza que la técnica de *Train-Test*.
- Entrenado y evaluado varias veces → más preciso.
- Indicado para algoritmos no tan lentos → procedimiento más lento.
- Elegir correctamente el número *k* → subconjuntos representativos.

2.4. Validación cruzada *Leave-One-Out*

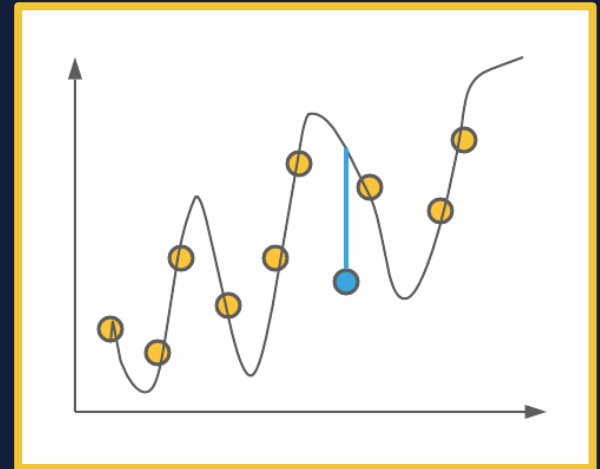
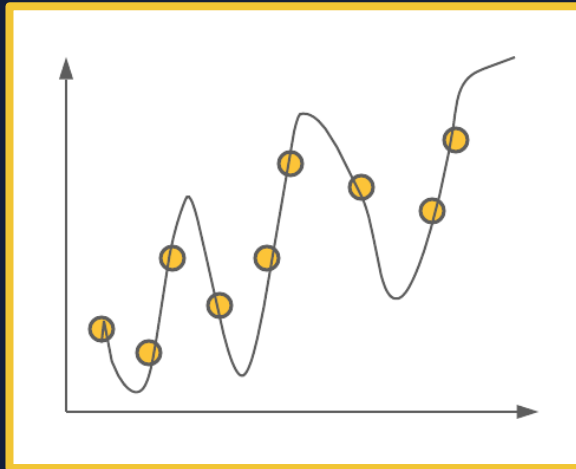
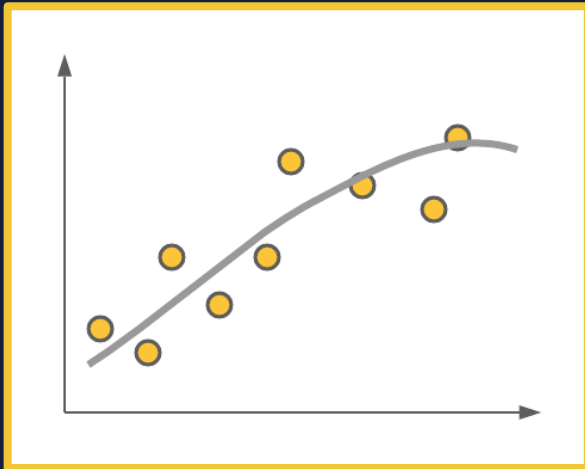
Si se configura la validación cruzada de forma que el tamaño de cada subconjunto es igual a 1, o lo que es lo mismo el número de *folds* es igual al número de patrones de la base de datos, estamos ante lo que se denomina la validación cruzada *Leave One Out*.

De esta forma, se obtienen k resultados, referentes a la evaluación del modelo obtenido que cada iteración, con el objetivo de estimar de una forma más razonable el rendimiento de dicho modelo. Sin embargo, tiene como desventaja que se trata de un procedimiento más costoso.

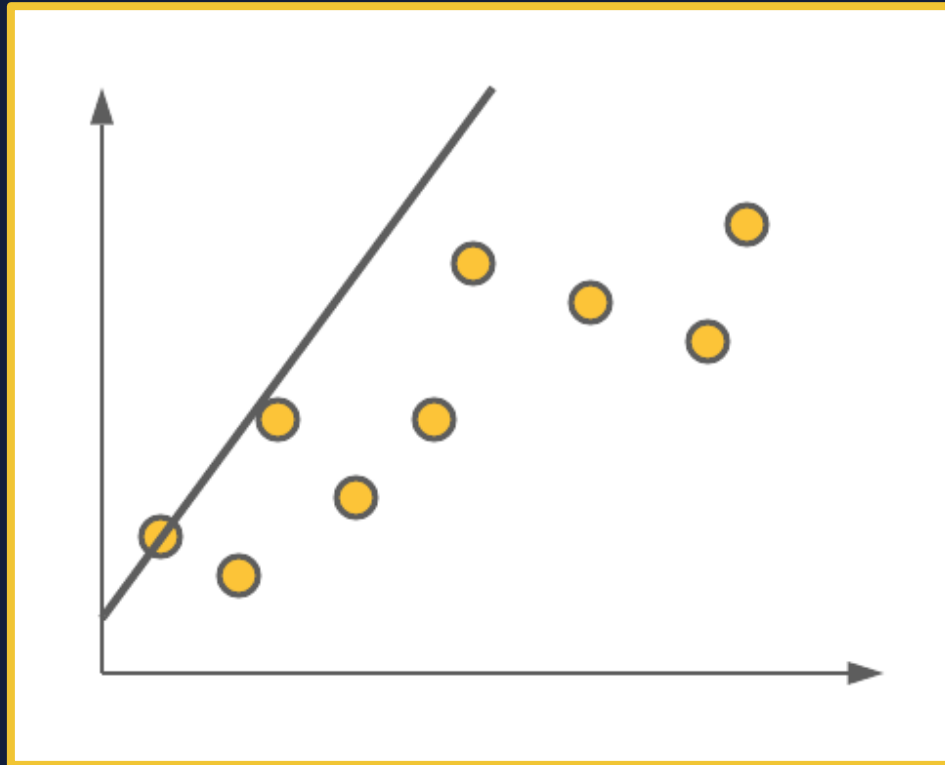
2.5. ¿Qué técnica es mejor?

- La validación cruzada *k-fold* es considerada la mejor forma de medir el rendimiento del algoritmo en el conjunto de test. Los valores más comunes de *k* son 3, 5 o 10.
- La división aleatoria *train-test* es rápida por lo que se recomienda su uso cuando estamos optimizando parámetros con algoritmos costosos y cuando se utilizan grandes bases de datos para reducir la varianza.
- El *Leave One Out* y la repetición de varias divisiones aleatorias *train-test* pueden ser útiles cuando se pretende equilibrar la varianza en el rendimiento estimado, la velocidad de entrenamiento del modelo y el tamaño del conjunto de datos.

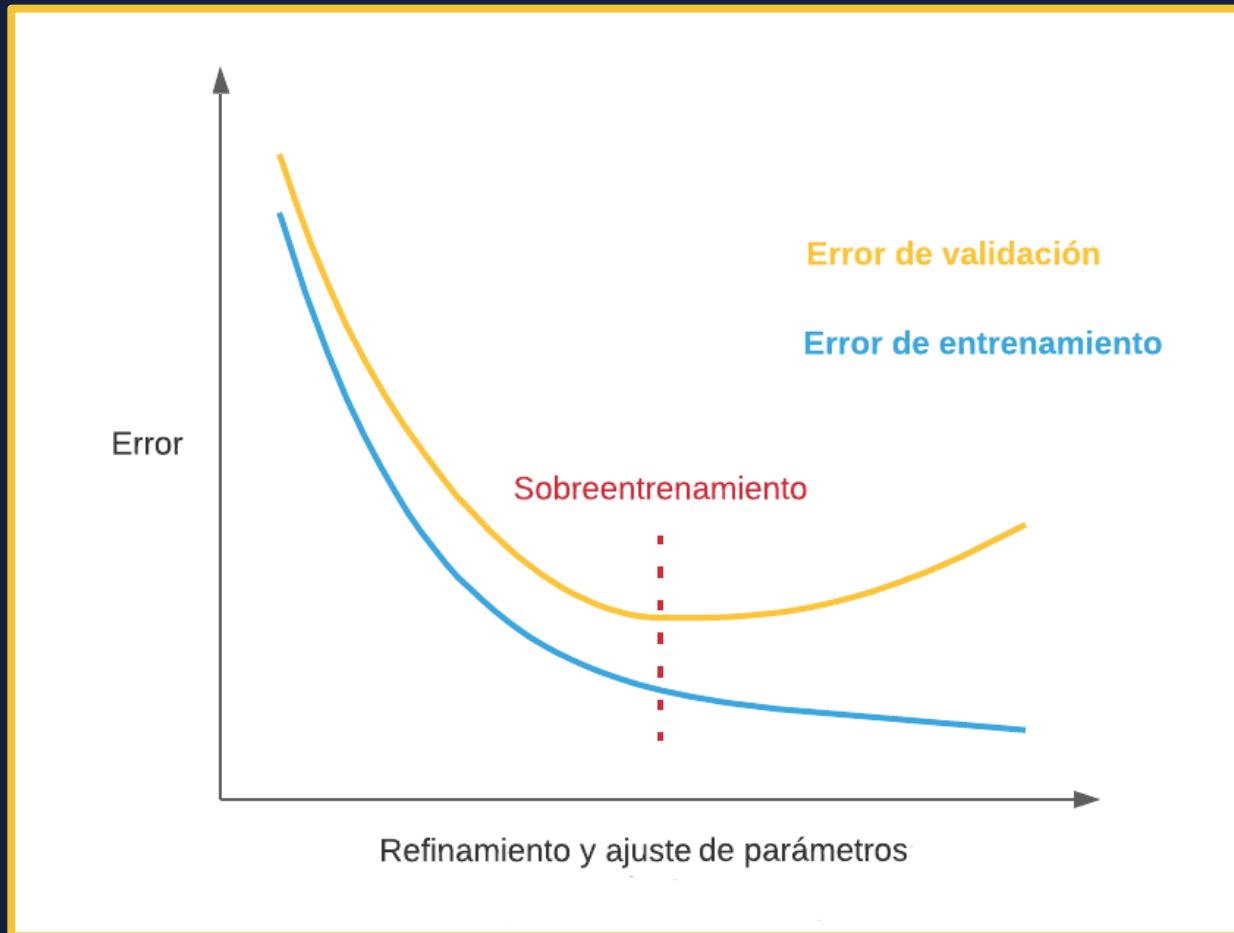
3.1. Sobreentrenamiento



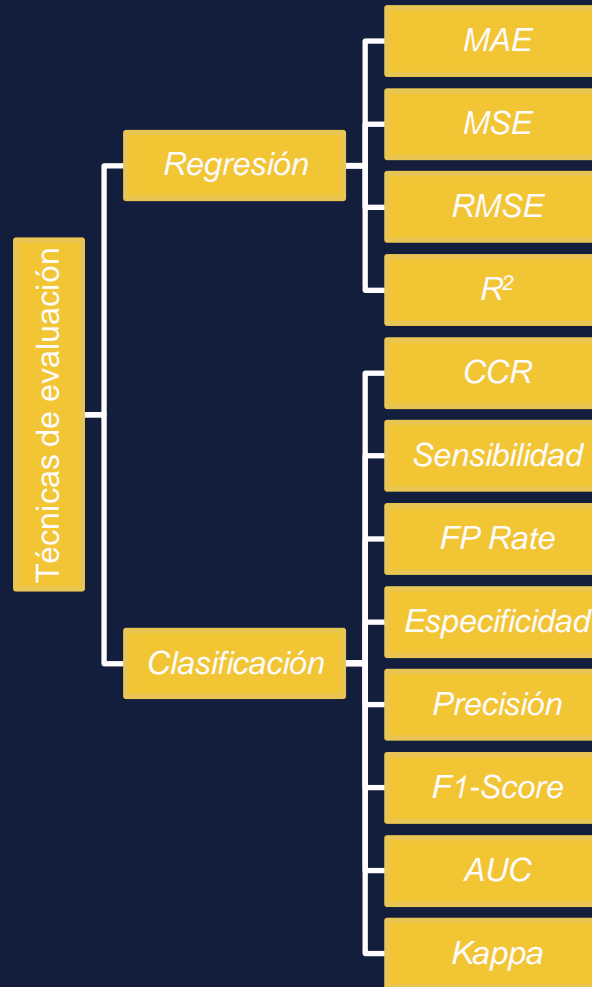
3.2. Infraentrenamiento



3.3. Equilibrio



4. Métricas de evaluación



4.1. Métricas de regresión

- **Error absoluto medio (*Mean Absolute Error, MAE*):** es la media de las diferencias absolutas entre el valor predicho y el real. Da una idea de cómo de erróneas fueron las predicciones, y de la magnitud de error pero no de su dirección. Un inconveniente es que no penaliza los grandes errores. Así, siendo n el número de patrones del conjunto, el MAE se define como:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **Error cuadrático medio (*Mean Squared Error, MSE*):** es la media de las diferencias de los errores al cuadrado. Es muy utilizado ya que es derivable y además, penaliza los errores más grandes. El MSE se define de la siguiente forma:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

4.1. Métricas de regresión

- **Raíz cuadrada del error cuadrático medio (*Root Mean Squared Error, RMSE*):** para que la medida del error esté en las mismas unidades que los valores de la variable objetivo, se aplica la raíz cuadrada al *MSE*.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- **Coeficiente de determinación, R^2 :** nos proporciona una medida de calidad del modelo para predecir los resultados. Está acotado entre 0 y 1, según no se ajuste o se ajuste a los datos, respectivamente. Se calcula de la siguiente forma:

$$R^2 = \frac{\sigma_{y,\hat{y}}^2}{\sigma_y^2 \sigma_{\hat{y}}^2}$$

4.2. Métricas de clasificación

- Comparar si el valor real coincide con el valor predicho.
- Para ello, usamos la **matriz de confusión**.

		Clase predicha	
		Clase positiva	Clase negativa
Clase real	Clase positiva	TP	FN
	Clase negativa	FP	TN

4.2. Métricas de clasificación

- Comparar si el valor real coincide con el valor predicho.
- Para ello, usamos la **matriz de confusión**.

Error tipo I (falso positivo)



Error tipo II (falso negativo)



4.2. Métricas de clasificación

- **Precisión global (*Accuracy* o *CCR*):** porcentaje de patrones correctamente clasificados:

$$CCR = \frac{TP + TN}{N}$$

- **Sensibilidad (*Recall* o *TP Rate*):** porcentaje de patrones positivos predichos como positivos:

$$Sensibilidad = \frac{TP}{TP + FN}$$

- **False Positive Rate (*FP Rate*):** porcentaje de patrones negativos predichos como positivos:

$$FP\ Rate = \frac{FP}{TN + FP}$$

4.2. Métricas de clasificación

- **Precisión:** número de patrones positivos predichos como positivos frente al total de patrones predichos como positivos:

$$Precisión = \frac{TP}{TP + FP}$$

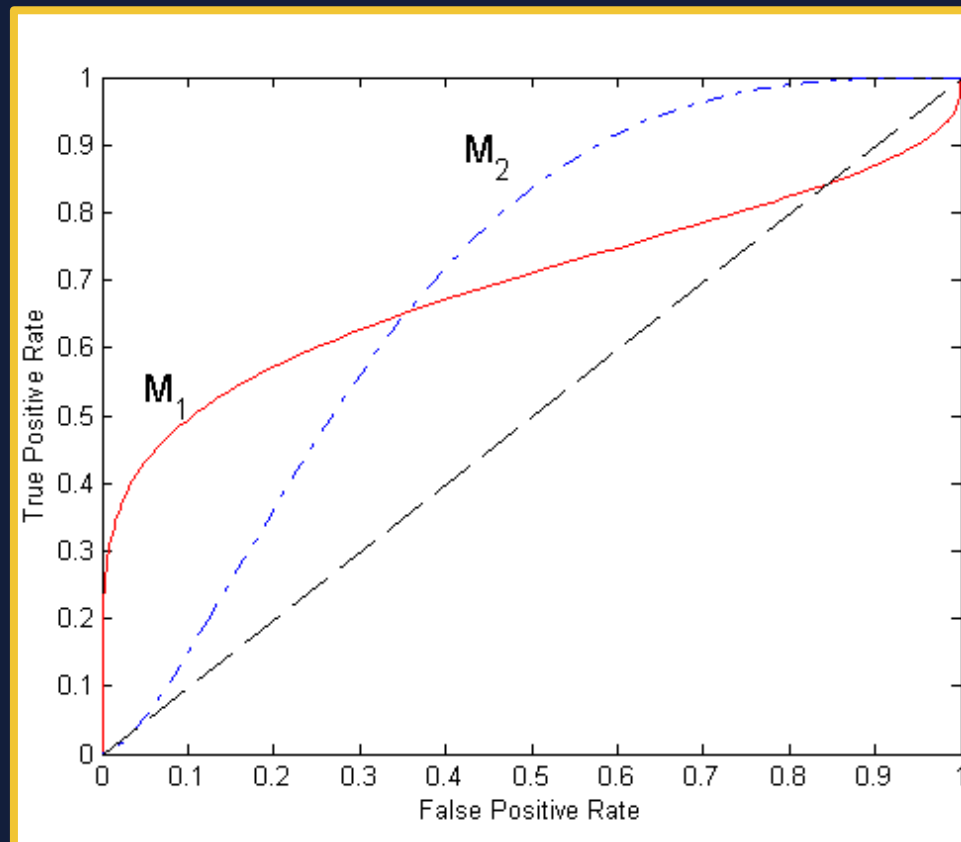
- **F1-Score (Recall o TP Rate):** relaciona las métricas de Sensibilidad y Precisión.

$$F1 - Score = \frac{2 * Precisión * Recall}{Precisión + Recall}$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN}$$

4.2. Métricas de clasificación

- **Área bajo la curva ROC:** caracteriza el compromiso entre aciertos y falsas alarmas.



4.2. Métricas de clasificación

En problemas multiclase, estas métricas se calculan considerando una clase frente a todas las demás. La clase seleccionada será considerada como la clase Positiva y el resto como la negativa. De este modo, obtendremos un valor de cada métrica para cada una de las clases.

4.2. Métricas de clasificación

- **Kappa:** mide el acuerdo o la relación entre los valores reales y los predichos.

$$Kappa = \frac{p_0 - p_e}{1 - p_e}$$

$$p_0 = CCR = \frac{1}{n} \sum_{j=1}^J n_{jj}$$

$$p_e = \frac{1}{n^2} \sum_{j=1}^J n_{j.} \cdot n_{.j}$$

MUCHAS GRACIAS POR SU ATENCIÓN



antoniomdr18@gmail.com



Antonio Manuel Durán Rosal

<https://www.linkedin.com/in/antonio-manuel-dur%C3%A1n-rosal-99ba92a5/>



twitter.com/eiposgrados



facebook.com/eiposgrados



instagram.com/eiposgrados